CLAIMS

1. An apparatus for performing commands, comprising:

performance logic, wherein the performance logic is at least configured to perform a plurality of commands issued by a processor, and wherein the performance logic further comprises a command queue with a predefined number of slots for storing the plurality of commands issued by the processor;

a command pipeline, wherein the command pipeline at least communicates the plurality of commands issued by the processor to the performance logic;

a plurality of counters, wherein the plurality of counters at least determine a known count of a number of commands in the command pipeline and in the command queue, and wherein the plurality of counters at least predicts an unknown count of future commands; and

stall logic, wherein the stall logic at least has the ability to stall performance of the plurality of commands issued by the processor based on at least a use of the unknown count and the known count.

2. The apparatus of Claim 1, wherein the performing logic further comprises:

fetch logic, wherein the fetch logic at least retrieves the plurality of commands from the command pipeline to at least provide a plurality of fetched command;

decode logic, wherein the decode logic at least decodes the plurality of fetched command to at least provide a plurality of decoded commands;

issue logic, wherein the issue logic at least issues the plurality of decoded commands to the command queue of the performance logic; and

execution logic to execute the plurality of decoded command in the command queue.

3. The apparatus of Claim 1, wherein the plurality of counters further comprises:

a known counter, wherein the known counter at least determined the known count; and

an unknown counter, wherein the unknown counter at least determines the unknown count.

4. The apparatus of Claim 3, wherein the stall logic further comprises a stall monitor, wherein the stall monitor at least stalls performance of the plurality of commands issued by the processor if a sum the unknown count and the known count is greater than the predefined number of slots in the command queue.

5. The apparatus of Claim 1 wherein the stall logic further comprises:

a tracking pipeline, wherein the tracking pipeline monitors progress of the plurality of commands and stall requests to at least provide a control signal;

an incrementer, wherein the incrementer increments the unknown count based on the control signal; and

a decrementer, wherein the unknown count based on the control signal.

6. The apparatus of Claim 4 wherein the stall logic further comprises:

a tracking pipeline, wherein the tracking pipeline monitors progress of the plurality of commands and stall requests to at least provide a control signal;

an incrementer, wherein the incrementer increments the unknown count based on the control signal; and

a decrementer, wherein the unknown count based on the control signal.

7. An apparatus for stalling command performance in a command performance system, comprising:

stall logic, wherein the stall logic at least has the ability to stall performance of a plurality of commands

issued by a processor based on at least a use of a known count of a number of commands in a command pipeline and in a command queue and an unknown count prediction of future commands.

8. The apparatus of Claim 7, wherein the stall logic further comprises:

a known counter, wherein the known counter at least determined the known count; and

an unknown counter, wherein the unknown counter at least determines the unknown count.

9. The apparatus of Claim 8, wherein the stall logic further comprises a stall monitor, wherein the stall monitor at least stalls performance of the plurality of commands issued by the processor if a sum the unknown count and the known count is greater than a predefined number of slots in the command queue.

10. The apparatus of Claim 7 wherein the stall logic further comprises:

a tracking pipeline, wherein the tracking pipeline monitors progress of the plurality of commands and stall requests to at least provide a control signal;

an incrementer, wherein the incrementer increments the unknown count based on the control signal; and

a decrementer, wherein the unknown count based on the control signal.

11. The apparatus of Claim 9 wherein the stall logic further comprises:

a tracking pipeline, wherein the tracking pipeline monitors progress of the plurality of commands and stall requests to at least provide a control signal;

an incrementer, wherein the incrementer increments the unknown count based on the control signal; and

a decrementer, wherein the unknown count based on the control signal.

12. The method of stalling performance of commands in a command performance system, comprising:

executing a plurality of commands;

reporting command progress of the plurality of commands to stall logic during execution;

determining if the performance misses during execution;

if the performance misses, storing the command in a command queue; and

stalling the command performance based on misses and progress of the plurality of commands.

13. The method of Claim 12, where the step of stalling the command performance further comprises receiving a completion signal when stored commands are performed.

14. The method of Claim 13, wherein the step of stalling the command performance further comprises:

determining a known count of a number of commands in a command pipeline and in a command queue; and

determining an unknown count prediction of future commands.

15. The method of Claim 14, wherein the step of stalling the command performance further comprises:

determining a sum of the known count and the unknown count; and

determining if the sum is greater than a predefined number of slots in the command queue; and

if the sum is greater than a predefined number of slots in the command queue, requesting a stall.

16. The computer program product for stalling performance of commands in a command performance system, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for executing a plurality of commands;

computer code for reporting command progress of the plurality of commands to stall logic during execution;

computer code for determining if the performance misses during execution;

if the performance misses, computer code for storing the command in a command queue; and

computer code for stalling the command performance based on misses and progress of the plurality of commands.

17. The computer program product of Claim 16, where the computer code for stalling the command performance further comprises receiving a completion signal when stored commands are performed.

18. The computer program product of Claim 17, wherein the computer code for stalling the command performance further comprises:

computer code for determining a known count of a number of commands in a command pipeline and in a command queue; and

computer code for determining an unknown count prediction of future commands.

19. The computer program product of Claim 18, wherein the computer code for stalling the command performance further comprises:

computer code for determining a sum of the known count and the unknown count; and

computer code for determining if the sum is greater than a predefined number of slots in the command queue; and

if the the sum is greater than a predefined number of slots in the command queue, computer code for requesting a stall.